2

WL-TR-93-1046

DETAILED EXAMPLE OF USING ACEC RELEASE 3.0 WITH THE DEC VAX ADA COMPILER

Major Steven A. Davidson
Software Concepts Group
Avionics Logistics Branch
Systems Avionics Division

December 1992

Final Report for Period 13 Jul 92 to 24 Jul 92

Approved for Public Release; Distribution Unlimited

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7409

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporations; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Steven A. Davidson

STEVEN A. DAVIDSON, Major
U.S. Air Force Reserves

Robert L. Harris

ROBERT L. HARRIS, Group Chief
Software Concepts Group
Avionics Logistics Branch

Donna M. Morris

DONNA M. MORRIS, Chief
Avionics Logistics Branch
Systems Avionics Division

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization, please notify WL/AAAF, Wright-Patterson AFB, OH 45433-7409 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
DEC 1992

3. REPORT TYPE AND DATES COVERED
FINAL 13 Jul 92 to 24 Jul 92

4. TITLE AND SUBTITLE
DETAILED EXAMPLE OF USING ACEC RELEASE 3.0 WITH THE DEC VAX ADA COMPILER

5. FUNDING NUMBERS
PE 63756D
PR 2853
TA 01
WU 03

6. AUTHOR(S)
Maj. Steven A. Davidson, USAF Reserves

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Avionics Directorate
Wright Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, Ohio 45433-7409

8. PERFORMING ORGANIZATION REPORT NUMBER
WL-TR-93-1046

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Avionics Directorate
Wright Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, Ohio 45433-7409

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-93-1046

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release;
distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report provides explicit, step-by-step instructions on how to execute the Ada Compiler Evaluation Capability (ACEC) software tool on a Digital Equipment Corp. VAX computer. It provides the reader a "worked example" of execution for using this tool more effectively. It covers version 3.0 release of the ACEC.

14. SUBJECT TERMS
Ada Compiler Evaluation Capability, Ada computer language, DEC VAX usage, Software Tools, ACEC

15. NUMBER OF PAGES
45

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19 SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

FOREWORD

This technical report was prepared by Major Steven A. Davidson, USAF Reserves, as part of his 2-week active-duty tour at Wright Laboratory, Avionics Directorate, System Avionics Division, Avionics Logistics Branch (WL/AAAF), Wright-Patterson AFB OH during the period 13-24 July 1992.

This is the third of a series of reports that document efforts using the Department of Defense Ada Compiler Evaluation Capability (ACEC) software tool residing at WL/AAAF on its VAX/4000 cluster. The intent of these reports is to provide an easier access to the ACEC for researchers within the Government by providing a "hands-on" experience guide.

The work was accomplished under the Aeronautical Systems Center Air Force Reserves project 90-432-LAB.

TABLE OF CONTENTS

Section 1 - INTRODUCTION

Section 2 - INITIAL SETUP

Section 3 - PRETEST STEPS

TABLE OF CONTENTS (CONT.)

Section 4 - RUNNING PERFORMANCE TESTS FOR A SUBGROUP

TABLE OF CONTENTS (CONT.)

Section 1
INTRODUCTION

1. Background.

The Ada Compiler Evaluation Capability (ACEC) is a software tool developed by Boeing Military Airplanes (BMA) for the Avionics Logistics Branch of the Avionics Directorate of Wright Laboratory (WL/AAAF) at Wright-Patterson Air Force Base, Dayton, Ohio. The ACEC measures the performance of an Ada compiler hosted on a specific computer. The ACEC Version 3.0 can compile and execute over 1600 Ada test routines which exercise different features of the Ada language compiler. Compilation speed, compiled code memory size, and execution time are measured. The ACEC user can use this information to compare performance between compiler/computer combinations, isolate strengths and weaknesses of a compiler/computer pair, and determine changes between releases of a compiler (1).

2. The ACEC User's Guide.

BMA wrote a User's Guide (2) to help the user initialize and execute the ACEC. The User's Guide for Version 3.0 gives detailed directions for installing the ACEC on a "generic computer." It also gives general guidance on executing the ACEC on a group of Ada performance tests programs.

3. Purpose.

The purpose of this paper is to provide an explicit, step-by-step worked-example of how Version 3.0 of the ACEC is installed on a Digital Equipment Corporation (DEC) VAX main frame computer.

Section 4 of this paper shows exactly how the "Avionics" subgroup in the "Applications" performance test group is executed on the VAX computer. Section 5 shows exactly what is done to execute the entire "Applications" performance test group.

4. Assumptions.

This paper makes the following assumptions:

a. User's Guide. The reader is expected to have a copy of the User's Guide for ACEC, as described in Reference 2. This paper makes frequent references to this document.

b. DEC Command Language (DCL). The reader is expected to be somewhat familiar with DCL commands used on VAX computers. All DCL commands necessary to obtain results are provided in this paper. However, users who are not comfortable with DCL are advised to obtain and refer to a DCL user's manual tu better understand the DCL commands used in this paper.

c. Ada Compilation System (acs,. The .COM files on the ACEC distribution tape are written such that the Ada compiler, linker, and library are invoked by means of the VAX DCL utility "acs". This utility software must be available on the VAX machine being used to compile the Ada test routines and the analysis routines (such as MEDIAN, or SSA).

d. Text File Editor. The reader is expected to be able to use an on-line text editor to modify or create VAX command files (.COM) written in ASCII text. The VAX text file screen editors such as EDT or the Language Sensitive Editor (LSE) are recommended. However, any text file editor (or word processor) capable of reading and writing ASCII text files will do.

e. Ada Compiler Language. The reader is not required to be familiar with the Ada Compiler Language to exercise ACEC using the steps listed in this paper.

5. Format Conventions.

This paper will use the following format conventions to promote understandability and minimize confusion:

a. Interactive On-Line Commands. Commands which are to be entered at an on-line VAX terminal are indented and set apart from the supporting text of each step by blank lines. Each command appears on a separate line. The end of the line implies the carriage return (RETURN) key is to be pressed.

b. File Names. File names discussed in the text will be enclosed within double-quote marks (example: "zc_setli.com").

c. Step Numbering. The numbering used for the "Steps" in Section 3 of this document are keyed to the numbering for the "Steps" in Appendix B of the User's Guide under the headings, "EXECUTING THE TEST:". This facilitates a direct correlation between Section 3 of this document and Appendix B of the User's Guide.

6. Flow Charts.

Figures 1 and 2 are flow charts that diagram the relationships among the Ada performance test group file (in these examples, "ap_av.com" or "ap.com"), the menu driven ACEC executive files (".exe") and the various intermediate data files, and the output report files produced by the ACEC. These flow charts also suggest the sequence of steps the user should follow to obtain results after the ACEC has been installed and pretested, as described in Section 3. Figure 1 directly relates to Section 4; Figure 2 directly relates to Section 5. Figures 1 and 2 are related to Figure 1 on Page 2.3 and Figure 10 on Page 8.11 of the User's Guide.

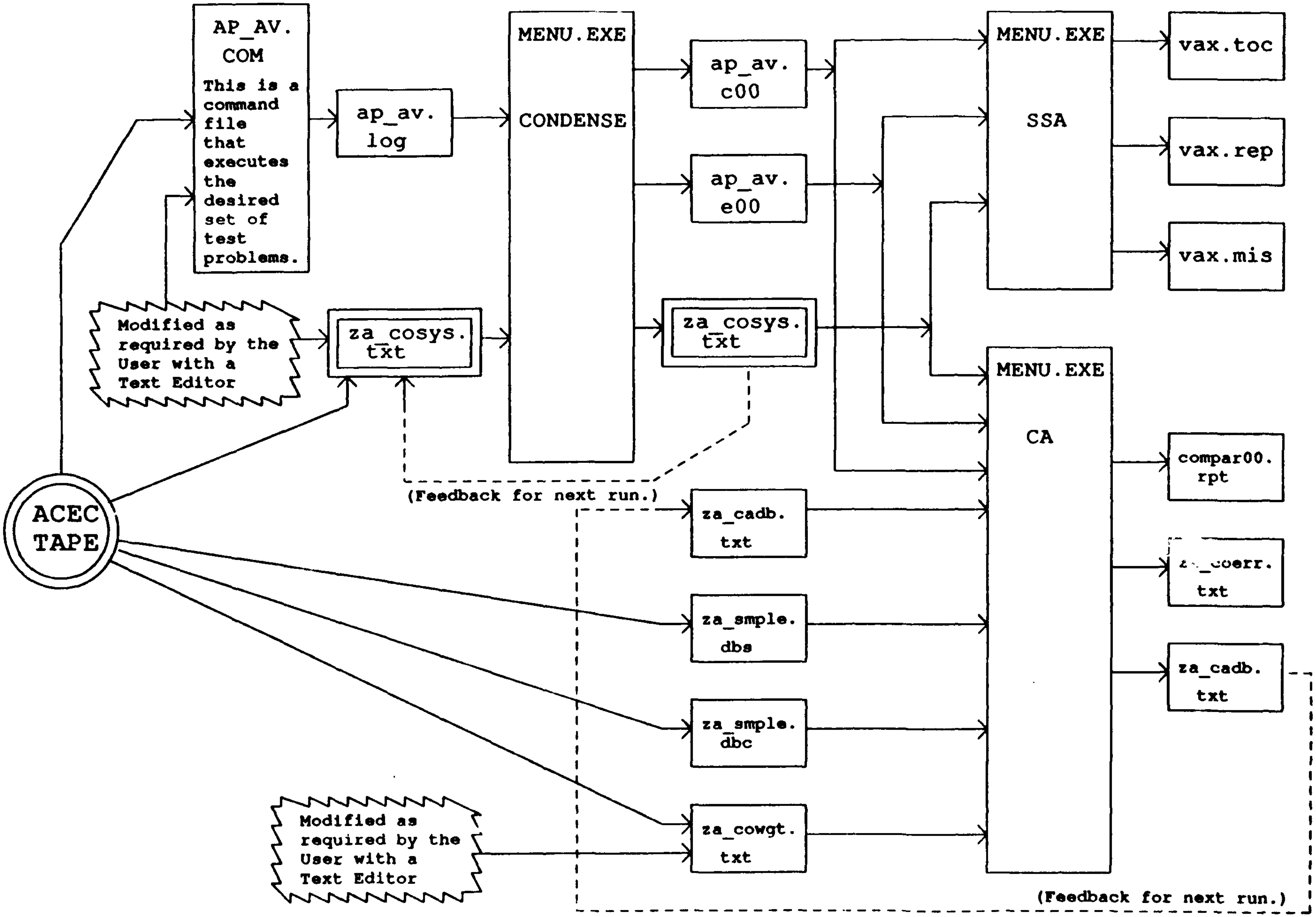

Figure 1. Running Performance Tests on the AVIONICS Subgroup

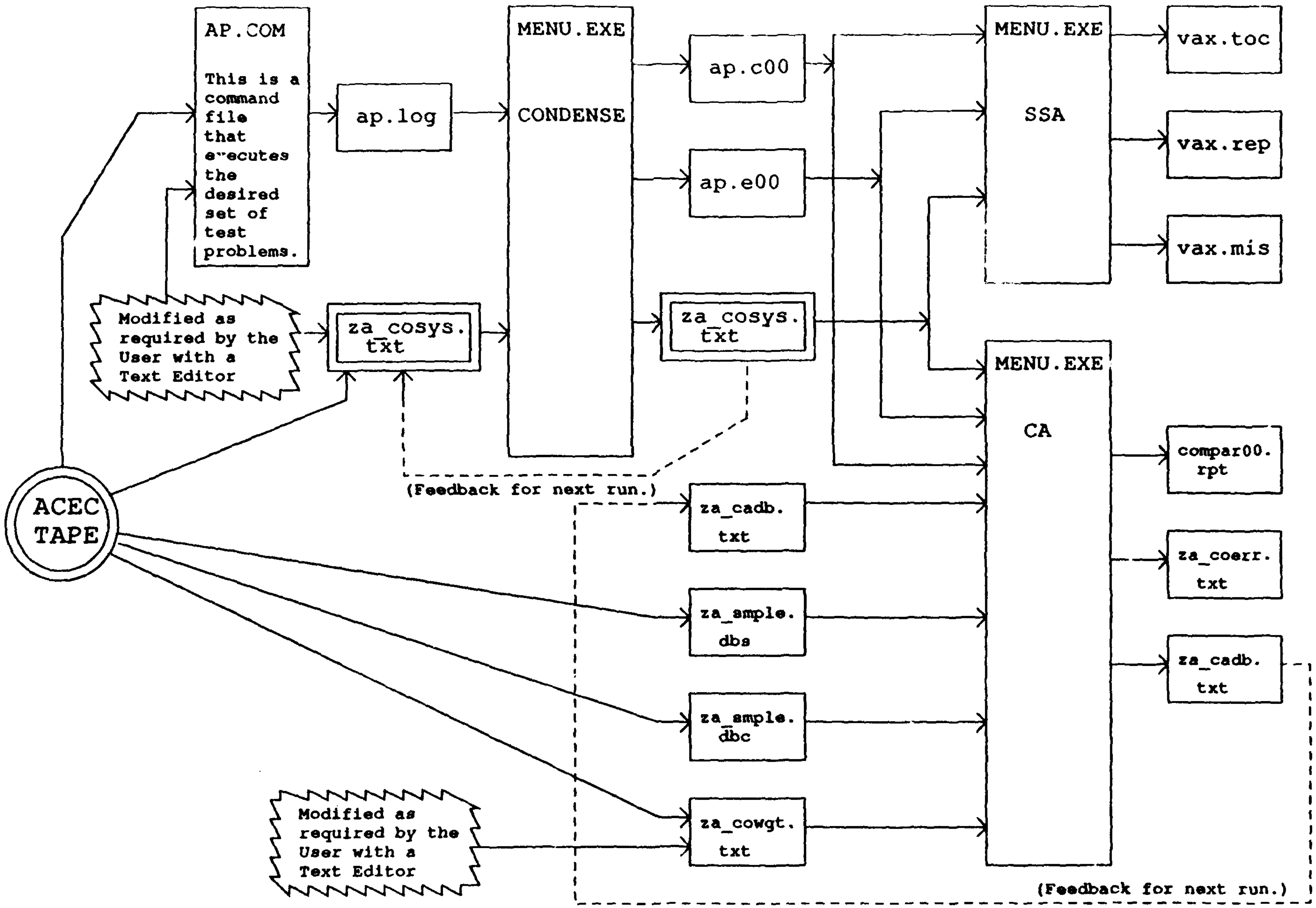

Figure 2. Running Performance Tests on the APPLICATION Group

Section 2

INITIAL SETUP

1. Step 1.01 - Read the ACEC Version 3 Distribution Tape

Read the ACEC Version 3 Distribution Tape into a public directory that has read-only protection. Request the help of your VAX System Manager to create the public directory and read the tape into it. The ACEC distribution tape was read into public directory $1$DIA9:[acec_v3] on the AAAF VAX.

2. Step 1.02 - Set Up the ACEC Work Directory

First, a structure of subdirectories should be set up in the user's area to contain the ACEC files. The steps in this paper for executing ACEC are based on the following directory and subdirectory structure. The top directory is [davidson]. Of course, the reader's VAX account name would be substituted for "davidson" in all the commands presented in this paper. The work subdirectory is set up by entering the VAX command:

```
create/dir [davidson.acec]
```

Subdirectory [davidson.acec] is the ACEC working directory. This is where all ACEC files are modified and executed.

3. Step 1.03 - Copy ACEC files Into the Work Directory

Copy the ACEC files needed from the public directory into the work directory. The ACEC Distribution Tape contains files for use with Unix systems. These are of no use on the VAX and therefore should not be copied into the work directory. All the ACEC files except the Unix files are copied into the work directory by using the VAX command:

```
copy/exclude=(*.unx) [acec_v3]*.* [davidson.acec]*.*
```

(You may have to specify Disk 9 for the public directory by using

```
copy/exclude=(*.unx) $1$DIA9:[acec_v3]*.* [davidson.acec]*.*
```

if your account is on another disk.)

Section 3

PRETEST STEPS

1. Appendix B of User's Manual

A significant improvement in Version 3 of the ACEC is the detailed step-by-step instructions provided for configuring the ACEC analysis programs on the user's computer. Section 5.2 of the User's Manual tells the User these detailed instructions are provided in file "zp_rdme1.txt" which can be printed and used as a hardcopy reference or displayed to another window if that capability is available on the host system. The entire contents of file "zp_rdme1.txt" is also contained in Appendix B of the User's Manual (which precludes the need to print a hardcopy of the file). This section of the paper will make frequent references to Appendix B. The steps numbered in this section also correspond to the steps numbered in Appendix B under the subheadings, "EXECUTING THE TEST:". Some parts of the Steps in Appendix B are skipped in this section (for example, modifying specific files) because they were not necessary for execution of the ACEC on the AAAF VAX computer.

2. Step 1.1 - Create the Ada Library for ACEC

The Ada library used by the VAX Ada compiler will be located in [davidson.acec.adalib] and MUST be created using the acs command by entering:

```
acs create library [davidson.acec.adalib]
```

(Do not attempt to create this directory using the VAX command similar to the one discussed in Step 1.02.)

3. Step 1.2 - Modify File "zc_setli.com"

Change the second line in this file from:

```
$ acs set library adalib
```

to:

```
$ ace set library [davidson.acec.adalib]
```

4. Step 1.3 - Modify File "zp_comp.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

This is the most frequent change to be made in modified files.

5. Step 1.4 - Execute Command File "zp_comp.com"

Enter the VAX command:

```
@zp_comp
```

Execution takes less than one minute to complete.

6. Step 1.5 - Modify File "zp_talk"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

7. Step 1.6 - Execute Command File "zp_talk.com"

Enter the VAX command:

```
@zp_talk
```

The purpose of this step is to verify the text input and output Ada package (TEXT_IO or equivalent) is working properly. The following results should appear on the screen:

```
Display floating point numbers  1.50000E+01   1.500E+1     15.0
Display floating point numbers -1.23450E+04  -1.235E+4 -12345.000
Display integers 100    22222
```

8. Step 2.1 - Modify File "zp_label.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

9. Step 2.2 - Execute the Test For Attribute label'ADDRESS

Enter the VAX command:

```
        @zp_label
```

Observe the results on the terminal screen. On the AAAF VAX, the following message appeared:

```
    The difference between two label'ADDRESSes is outside the
    expected range of 4 to 16 bytes.  Please compare this result
    with a system map or machine code listing to verify accuracy.
    SIZE IN 8 BIT BYTES =   0
```

This means that DEC Ada (Version V2.2-38 and below) does not support the standard label'ADDRESS attribute in Ada. It always returns a value of zero. Unless this changes in an upgraded version of DEC Ada, users on the AAAF VAX must use the assembly routine in file "zg_getadr.mar" to obtain code expansion size measurements. No modifications of file "zg_getadr.mar" are required.

10. Step 3.1 - Decide on CPU or Elapsed Time Measurements

Section 5.3.3 discusses the choice between CPU time measurements and elapsed time or "wall clock" measurements. The ACEC is set up to measure elapsed time as a default. Elapsed time measurements are the appropriate metric to collect on bare machine targets; that is, machines which are not time-shared and have no contending jobs. I initially selected the elapsed time measurement option with the AAAF VAX as the target machine. Because the AAAF VAX is time-shared by many users, results from execution time varied widely which caused ACEC to reject most of my execution time measurements as invalid in the final statistical analysis reports. Therefore, since it is most likely the user will initially want to learn ACEC using the main-frame VAX, the CPU time measurements should be selected.

After the VAX user is comfortable using the ACEC and a target machine (which is not time shared) is ready to be tested, all of the Pretest Steps (except possibly Steps 12 through 14, compilation and testing of the ACEC analysis tools) should be performed on the target machine, selecting options for elapsed time instead of CPU time.

11. Step 3.3 - Modify File "zp_tcalc.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

For CPU time measurements, no other modifications to this file should be made. All four "zp_tcal*" programs will be compiled and linked by this command file.

12. Step 3.4 - Execute Command File "zp_tcalc.com"

Enter the VAX command:

```
@zp_tcalc
```

Execution takes about 2 minutes.

13. Step 4.1 - Execute Timing Program "zp_tcal1"

This program MUST be executed interactively from the computer terminal. A stopwatch is preferred but access to at least a regular watch or wall clock is necessary for this test. Execute this test by entering the VAX command:

```
run zp_tcal1
```

14. Step 4.2 - Verifying Test Results

Instructional text will appear on the screen. Then the line:

```
TIMING WILL BEGIN IN     23 SECONDS
```

will appear. The number of seconds depends on when the computer system clock will be as 00 seconds. Thereafter, 15 lines will be printed to the screen, with exactly a 1-minute delay between each line. An example of the first few lines are as follows:

```
number of minutes since start is        0

number of minutes since start is        1

number of minutes since start is        2

          .
          .
          .
```

The program will run for 15 minutes until all 15 lines are printed. The accuracy of the DEC Ada routine CALENDAR.CLOCK on the VAX is measured by comparing the difference between the stopwatch time and the time required to print the 15 statement lines. The time difference measured on the AAAF VAX over 15 minutes was less than 1 second which responds to a 0.11 percent error or an accuracy of 99.89 percent.

If the target machine is not the VAX, then "zp_tcalc.com" and "zp_tcal*" programs must be run on the target machines to determine the accuracy of the timing measurements.

Although the test can be interrupted prematurely to perform some VAX commands by entering "Ctrl-y" for "interrupt," I was unable to execute any other program, including the text editor, until "zp_tcal1" completed execution. The "Ctrl-y" also suspended execution of "zp_tcal1" until execution of another program was attempted. The point is, if you are in a hurry, do not start these timing programs. Once started, they will prevent you from running other programs for at least 15 minutes.

15. Step 4.3 - Execute Timing Program "zp_tcal2"

This timing program utilizes a different method of creating the one-minute delay between statements. However, operation and output are exactly the same for this program as for "zp_tcal1" (refer to Steps 4.1 and 4.2 above).

16. Executing Timing Program "zp_tcal3"

This test requires that it be run on the target machine both when it is running contending tasks and when it has no contending tasks. Because it was not possible to execute on the AAAF VAX and prevent all other contending tasks (from other users) from executing, this test was not attempted. Later successful analysis results indicate results from the "zp_tcal3" would have been satisfactory.

17. Executing Timing Programs "zp_tcal4" and "zp_tcal5"

These programs determine whether process time or program time is being measured for compilation CPU time. Process time is desired. (Refer to Section 5.3.4 of the User's Manual.) File "zp_tcalr.com" must be modified. Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

Submit the command file as a batch job by entering the VAX command:

```
  submit/notify/log_file=[DAVIDSON.ACEC]zp_tcalr.log zp_tcalr.com
```

This file takes about 5 minutes to execute. The results will be written to file "zp_tcalr.log" in the [DAVIDSON.ACEC] directory. This file can be viewed with an editor and printed. Results of this test on the AAAF VAX show the CPU time is process time, which is the desired result.

18. Step 5.1 - Modify File "zg_basln.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

No other modifications to file "zg_basln.com" are required.

19. Step 5.1a - Modify File "zg_basel.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

The lines to compile "zg_glob1.ada" and "zg_glob2.ada" can be commented out as these programs were compiled by the command file "zp_comp.com" in Step 1.4. However, there is no harm in leaving this part of the file unchanged and recompiling these programs.

The default setting is for "Option 1 - ZG_GLOB3.ELG" which is to use elapsed time for timing measurements and "zg_getadr.mar" for code size measurements. As discussed previously, if the AAAF VAX is the target machine, "Option 3 - ZG_GLOB3.CPG" should be selected to use CPU time in place of elapsed time with "zg_getadr.mar" for code size measurements. This is done by modifying the statements under Option 1 to comment out these statements by inserting the "!" symbol immediately after the "$" symbol for all the command lines under Option 1. The statements under Option 3 are uncommented by deleting the "!" symbol immediately after the "$" symbol at the beginning of each of the command lines under Option 3. For example, the statements under Option 1 will be modified to:

```
$!       OPTION 1  -  ZG_GLOB3.ELG
$! set noon
$! acs delete unit zg_getad
$! macro zg_getad
$! @zc_adaop.com ZG_GLOB3.ELG
$! acs enter foreign zg_getad.obj zg_getad
$! set on
```

(The comments to the right of each command line which also begin with the comment symbol "!" are not shown above for space reasons but should be left unchanged.)

Similarly, the statements under Option 3 will be modified to:

```
$!       OPTION 3  -  ZG_GLOB3.CPG
$!! Adapt ZG_CPU.DEC to target compilation system and
   compile it (DEC EXAMPLE)
$ @zc_adaop.com zg_cpu.ada
$ set noon
$ acs delete unit zg_getad
$ macro zg_getad
$ @zc_adaop.com ZG_GLOB3.CPG
$ acs enter foreign zg_getad.obj zg_getad
$ set on
```

(The comments to the right of each command line which also begin with the comment symbol "!" are not shown above for space reasons but should be left unchanged.)

As the comment under Option 3 warns, this option will look for a file called "zg_cpu.ada" to compile. File "zg_cpu.ada" is not provided on the ACEC Distribution Tape. However, file "zg_cpu.dec" is provided as a suggested version for "zg_cpu.ada" if the target machine runs DEC Ada. When the AAAF VAX is the target machine, the required "zg_cpu.ada" file is created by simply copying file "zg_cpu.dec" to file "zg_cpu.ada" by entering the following VAX command (which should be done at this time):

```
    copy zg_cpu.dec zg_cpu.ada
```

No other modifications to file "zg_cpu.ada" or "zg_base1.com" are required.

20. Step 5.1b - Modify File "zg_base2.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

The default setting is the 32-bit random generator. No additional modifications to file "zg_base2.com" are required.

21. Step 5.1c - Modify File "zg_base3.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

The default setting is for Option 3, ZM_MATH using portable ZM_GENMA. This option works for the AAAF VAX as was demonstrated by the successful testing using this math package in Step 7 (explained later).

22. Step 5.1d - Modify File "zg_base4.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

Use the "substitute" command in your text editor to change the extensions for file "zg_init" from ".#" to ".CPG" which corresponds to Option 3 selected in Step 5.1a. There are five occurrences of this change which must be made in file "zg_base4.com". These changes should occur in Lines 18, 29, 40, 51, and 62 in file "zg_base4.com". No other changes to file "zg_base4.com" are required.

23. Step 5.1e - Modify File "zg_base5.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

Change the file extension of file "zg_incld" in Line 20 from ".#" to ".CPG" which corresponds to Option 3 selected in Step 5.1a. No other changes to file "zg_base5.com" are required.

24. Step 5.1f - Modify File "zg_base6.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

The default setting is for Option 1, elapsed time. This must be changed to Option 2 for CPU time, to correspond with Option 3 selected in Step 5.1a. Comment out all the command lines under Option 1 (Lines 15 through 18) by inserting the comment symbol "!" immediately after the "$" symbol, as was done in Step 5.1a. Uncomment the command lines under Option 2 (Lines 21 through 24) by deleting the comment symbol "!" immediately after the "$" symbol, as was done in Step 5.1a. There are no other changes required for file "zg_basic6.com".

25. Step 5.2 - Execute File "zg_basln.com"

Submit file "zg_basln.com" as a batch job because of the rather lengthy execution time and have the output written to a file, "zg_basln.log". Enter the following VAX command:

```
  submit/notify/log_file=[davidson.acec]zg_basln.log zg_basln.com
```

Execution should take approximately 10 minutes. The following files should be created as a result of this batch job:

```
          zg_cpy.ck         zg_incld.exe
          zg_cpy.db         zg_time1.exe
          zg_cpy.no         zg_time2.exe
          zg_cpy.op         zg_link1.exe
          zg_cpy.sp         zg_link2.exe
```

26. Step 6.1 - Modify File "zp_basic.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

Change the file extension of file "zp_basic" in Line 15 from ".#" to ".CPG" which corresponds to Option 3 selected in Step 5.1a. No other changes to file "zg_base5.com" are *required.*

27. Step 6.2 - Execute File "zp_basic.com"

Submit file "zp_basic.com" as a batch job because of the VERY lengthy execution time and have the output written to a file, "zp_basic.log". Enter the following VAX command:

```
  submit/notify/log_file=[davidson.acec]zp_basic.log zp_basic.com
```

Execution should take approximately 30 minutes. Print the resulting file "zp_basic.log" and review it. The default value for BASIC_ITERATION_COUNT is 2**17-1. When this step was performed on the AAAF VAX, the text printed in the output file "zp_basic.log" recommended the value for BASIC_ITERATION_COUNT be no less than 2**21-1 so that the value for BASIC_ITERATION_COUNT is past the point of convergence where the variations in the null loop time are less than 1 percent (see Page B21 in Appendix B of the User's Manual). The value of BASIC_ITERATION_COUNT is not set directly but is determined by the values for the variables BASIC_ITERATION_COUNT_LOWER and BASIC_ITERATION_COUNT_UPPER. The contents of file "zp_basic.log" also shows the values to use for these two independent variables to determine the recommended value for BASIC_ITERATION_COUNT by providing the following relationships:

```
    For BASIC_ITERATION_COUNT equal 2**17-1
      BASIC_ITERATION_COUNT_LOWER  =  16383
      BASIC_ITERATION_COUNT_UPPER  =      7
```

and also:

```
    For BASIC_ITERATION_COUNT equal 2**21-1
      BASIC_ITERATION_COUNT_LOWER  =  16383
     BASIC_ITERATION_COUNT_UPPER  =    127
```

These variables are set in the files "zg_glob3.elg", "zg_glob3.ell", "zg_glob3.cpg", and "zg_glob3.cpl".

28. Step 6.5 - Modify File "zg_glob3.cpg"

File "zg_globe.cpg" must be modified because this is the "zg_glob3.*" file used, based on Option 3 selected in Step 5.1a. Viewing file "zg_glob3.cpg" with a text editor, the variable BASIC_ITERATION_COUNT_LOWER is assigned in Line 44 the value of constant TWO_TO_14_MINUS_ONE which is previously set in Line 41 to the value of 2**14-1 which is equal to 16383. This is the value recommended for BASIC_ITERATION_COUNT_LOWER by executing "zp_basic.com" in Step 6.2. Therefore, no action is required for this variable.

Variable BASIC_ITERATION_COUNT_UPPER is initially assigned a value of 7 in Line 45. This corresponds to the default setting for BASIC_ITERATION_COUNT of 2**17-1 (see Step 6.2 discussion). To change BASIC_ITERATION_COUNT to the recommended value of 2**21-1, BASIC_ITERATION_COUNT_UPPER should be set equal to 127. Therefore, Line 45 in file "zg_glob3.cpg" was modified to:

```
  basic_iteration_count_upper : integer16 := 127;
```

No other changes to file "zg_glob3.cpg" are required.

29. Step 6.6 - Recompile the Package Body "zg_glob3"

Enter the following VAX commands:

```
set default [davidson.acec]
acs set library [.adalib]
acs load zg_glob3.cpg
acs compile/list zg_glob3
```

Allow the VAX system time to indicate each of the "acs" commands was completed before entering the next "acs" command.

Check all extensions of file "zg_glob3.log" created by this compilation for any error messages; two extensions were created when I ran a single compilation on the AAAF VAX.

If compilation errors are found, review of the files "zg_glob3.lis;*" may help resolve the problem. Check all extensions of "zg_glob3.lis" if there are problems; four extensions of this file were created when I ran a single compilation on the AAAF VAX.

30. Step 7.1 - Modify File "zp_dptst.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

No other changes to file "zp_dptst.com" are necessary.

31. Step 7.2 - Execute File "zp_dptst.com"

Execute file "zp_dptst.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]zp_dptst.log zp_dptst.com
```

This batch job will complete execution in less than 1 minute.

32. Step 7.3 - Examine the Results of "zp_dptst.com"

The results of Step 7.2 will be in file "zp_dptst.log". View file "zp_dptst.log" with an editor and print it. Check for any error messages flagged by "<<<ERROR>>>". By selecting the default Option 3 in file "zg_base3.com" (see Step 5.1c discussion), no errors were found when "zp_dptst.com" was executed on the AAAF VAX.

33. Step 8.1 - Modify File "zp_mttst.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

No other modifications to this file are required.

34. Step 8.3 - Execute File "zp_mttst.com"

Execute file "zp_mttst.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
submit/notify/log_file=[davidson.acec]zp_mttst.log zp_mttst.com
```

This batch job will complete execution in less than one minute.

35. Step 8.4 - Examine the Results of "zp_mttst.com"

The results of Step 8.3 will be in file "zp_mttst.log". View file "zp_mttst.log" with an editor and print it. Check for any error messages flagged by ">>> error". When "zp_mttst.com" was executed on the AAAF VAX, only one error occurred because "cos(zero)" did not equal exactly "one" contrary to NUMWG recommendations. As a result, "zp_mttst.com" showed:

```
cos(zero) - one is 1.19209E-07
```

36. Step 9.1 - Modify File "zp_dmtst.com"

Change the first line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

No other modifications to this file are required.

37. Step 9.3 - Execute File "zp_dmtst.com"

Execute file "zp_dmtst.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]zp_dmtst.log zp_dmtst.com
```

This batch job will complete execution in less than one minute.

38. Step 9.4 - Examine the Results of "zp_dmtst.com"

The results of Step 9.3 will be in file "zp_dmtst.log". View file "zp_dmtst.log" with an editor and print it. Check for any error messages flagged by ">>> error". When "zp_dmtst.com" was executed on the AAAF VAX, only one error occurred because "cos(zero)" did not equal exactly "one" contrary to NUMWG recommendations. As a result, "zp_dmtst.com" showed:

```
    cos(zero) - one is  -1.11022302E-16
```

39. Step 10.1 - Modify File "zp_intst.com"

Change the first line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

Change the extension of file "zp_test" in Line 17 from ".#" to ".CPG" to correspond with Option 3 selected in Step 5.1a.

Uncomment Line 24 by deleting the comment symbol "!" in Column 2.

Change the extension of files "zp_test" and "zp_intst" in Line 24 from ".#" to ".CPG" to correspond with Option 3 selected in Step 5.1a.

Print file "zp_intst.inc" for later reference.

40. Step 10.2 - Execute "zp_intst.com"

Execute file "zp intst.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]zp_intst.log zp_intst.com
```

This batch job will complete execution in a few seconds.

41. Step 10.3 - Examine the Results of "zp_intst.com"

Compare file "zp_intst.inc" (before 'stoptime2' is included) against file "zp_test.cpg" (after 'stoptime2' lines are included). Verify additional lines were written into file "zp_test.cpg" in place of:

```
    PRAGMA include ("stoptime2");
```

Print file "zp_intst.log" (or view it with an editor) to see the results of the VAX command DIFFERENCES used to compare file "zp_test.cpg" with "zp_intst.cpg". The number of difference sections and records should be zero.

Results should show there are zero differences between sections and records. However, results show 2 sections and 6 records are different in "zp_intst.cpg". Visual examination of "zp_test.cpg" and "zp_intst.cpg" show the two files are nearly the same. Apparently, file "zp_intst.cpg" is not exactly correct. A few statements are in the wrong order in this file.

Files "zp_intst.cpg" and "zp_test.cpg" can also be visually compared to confirm they are the same.

42. Step 11.1 - Modify File "zp_cl_so.com"

Change the fifth line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

No other modifications to file "zp_cl_so.com" are required.

43. Step 11.3 - Execute File "zp_cl_so.com"

Step 11.2 in the User's Guide recommends command files contained in "zp_cl_so.com" be modified for the correct default directory. This step can be skipped because there are no command files contained in "zp_cl_so.com" or listed under Step 11 of the User's Guide which require this change (assuming all previous steps have been accomplished).

Execute file "zp_cl_so.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]zp_cl_so.log zp_cl_so.com
```

This batch job will complete execution in less than six minutes. Output file "zp_cl_so.log" is used as an input file in Step 13.

44. Step 12.1 - Modify File "zc_anlys.com" To Compile Analysis Tools

Change the third line of this file from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

Option 1, the default, should be selected. This is recommended by the User's Manual on Page B38 in Appendix B. Therefore, no additional changes to file "zc_anlys.com" are required.

45. Step 12.2 - Modify the Command Files Used By "zc_anlys.com"

Change the sixth line of the command files from:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

This must be done for the following five command files:

```
file "za_ca.com"
file "za_cn.com"
file "za_co.com"
file "za_mn.com"
file "za_sa.com"
```

46. Step 12.3 - Execute File "zc_anlys.com"

Execute file "zc_anlys.com" as a batch file with the results written to an output file which can be viewed with an editor and printed. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]zc_anlys.log zc_anlys.com
```

This batch job will complete execution in about 23 minutes.

However, Option 1 requires the user's VAX account be authorized for a very high byte count quota. Initially, I could get NONE of the Options in "zc_anlys.com" to function properly with the AAAF VAX System default byte count quota. The following error message would appear in the output file "zc_anlys.log":

```
    -RMS-E-ACC, ACP file access failed
    -SYSTEM-F-EXBYTLM, exceeded byte count quota
    %ACS-E-CL_ERRDURLIN, Error during ACS LINK operation
```

After I showed this error message to the AAAF VAX System Manager, he increased my byte count quota to 14000. I was then able to execute "zc_anlys.com" using Option 1 with no problem. The Reader may also need to have a similar adjustment made to his or her account for a successful execution of "zc_anlys.com".

47. Step 13.1 - Review File "za_cosys.txt" for the Condense Program

Review file "za_cosys.txt" with a text editor and verify it contains the following lines of text (starting in Column 1):

```
>system_name                     := SYSTEM1
>execution_log                   := zp_cl_so.log
>execution_condensed             := zp_cl_so.dbs
>compilation_log                 := zp_cl_so.log
>compilation_condensed           := zp_cl_so.dbc
```

This should be the initial contents of file "za_cosys.txt" from the ACEC Distribution Tape; no modifications should be necessary. (There are also additional systems listed in this file which are not needed in Step 13 but will be needed in Step 14.)

48. Step 13.2 - Run the Condense Program for SYSTEM1

Based on the instructions provided in file "za_cosys.txt" discussed in Step 13.1, the Condense Program will read file "zp_cl_so.log", created in Step 11.3, reformat the information, and write the condensed data base files "zp_cl_so.dbs" and "zp_cl_so.dbc" as output files. The Condense Program in Version 3 of the ACEC performs a function similar to the FORMAT Program in Version 2 of the ACEC. Execute the Condense Program of the ACEC by entering the VAX command:

```
run menu
```

The Main Menu should be displayed. Follow the instructions provided in Step 13 on Page B40 of Appendix B of the User's Guide, repeated here for convenience. Enter:

```
     a,next          <--- to choose CONDENSE
     a,next          <--- to choose SYSTEM1
     b,next          <--- to choose EXCEPTIONAL_DATA_REPORT
     a,do            <--- to run Condense immediately
     quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
          file "zp_cl_so.dbc"
          file "zp_cl_so.dbs"
          file "system1.exc"
          file "za_coerr.txt"
```

A new version of file "za_cosys.txt" will also be created. Verify the format (not content) of file "zp_cl_so.dbs" appears correct by comparing it to file "zp_sys_1.dbs". (There is no corresponding comparison file provided on the ACEC Distribution Tape to check the format of file "zp_cl_so.dbc".)

49. Step 14.1 - Review File "za_cosys.txt" for Comparative Analysis (CA)

Review file "za_cosys.txt" with a text editor and verify it contains the following lines of text (starting in Column 1):

```
    >system_name                          := sys_1
    >execution_log                        :=
    >execution_condensed                  := zp_sys_1.dbs
    >compilation_log                      :=
    >compilation_condensed                :=

    >system_name                          := sys_2
    >execution_log                        :=
    >execution_condensed                  := zp_sys_2.dbs
    >compilation_log                      :=
    >compilation_condensed                :=

    >system_name                          := sys_3
    >execution_log                        :=
    >execution_condensed                  := zp_sys_3.dbs
    >compilation_log                      :=
    >compilation_condensed                :=
>system_name                      := Sample
>execution_log                    :=
>execution_condensed              := za_smple.dbs
>compilation_log                  :=
>compilation_condensed            := za_smple.dbc
```

This should be the initial contents of file "za_cosys.txt" from the ACEC Distribution Tape; no modifications should be necessary.

50. Step 14.2 - Run the Comparative Analysis (CA) Program

Based on the instructions provided in file "za_cosys.txt" discussed in Step 13.1, the Comparative Analysis Program will read files the following condensed data base files for execution times which are provided on the ACEC Distribution Tape:

```
file "zp_sys_1.dbs"
file "zp_sys_2.dbs"
file "zp_sys_3.dbs"
```

The Reader should notice that because Step 14 uses "canned data" base files provided on the ACEC Distribution Tape, results verify the operation of ONLY the Comparative Analysis Program. Step 14 is completely independent of the Condense Program test in Step 13. Also, successful completion of Step 14 does NOT mean that ANY of the performance tests executed properly on the target machine.

The Comparative Analysis Program will produce analysis reports and write them to output files "arithm00.tim" for the Arithmetic performance test group and "classi00.tim" for the Classical performance test group, based on options selected while executing the Menu Program as explained below. Execute the Comparative Analysis Program of the ACEC by entering the VAX command:

```
run menu
```

The Main Menu should be displayed. Follow the instructions provided in Step 14 on Pages B42 and B43 of Appendix B of the User's Guide, repeated here for convenience. Enter:

```
b,next          <--- to choose COMPARATIVE_ANALYSIS
b,c,d,next      <--- to choose sys_1, sys_2, & sys_3
a,next          <--- to choose EXECUTION_TIME
a,b,next        <--- to choose APPLICATION and CLASSICAL
b,next          <--- to choose FULL_REPORT & SUMMARY_REPORT
a,do            <--- to run the analysis immediately
quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

file "arithm00.tim"
file "classi00.tim"
file "za_cadb.txt"

Verify the format and content of these first two files are correct by comparing the following files:

compare file "arithm00.tim" with file "za_caari.tim"
compare file "classi00.tim" with file "za_cacla.tim"

This can be done by visual inspection or by using the VAX Command DIFFERENCES by entering the commands:

```
differences/output=arith.dif arithm00.tim za_caari.tim
differences/output=class.dif classi00.tim za_cacla.tim
```

Then print or view the output files "arith.dif" and "class.dif". The only difference should be that the dates in the footer line of each page of the files are different. When this was done on the AAAF VAX, the only differences were the dates in the footer lines.

Section 4

RUNNING PERFORMANCE TESTS FOR A SUBGROUP

1. Step 1 - Modify a Performance Group Command File

The user may wish to perform analyses on a subset of the Performance tests to study the effects of selected types of problems on the target machines and to minimize computer processing time between analysis report outputs. This is especially desired when one is learning to use the ACEC.

Version 3 of the ACEC organizes over 1600 individual performance tests into 17 major performance test group command files, based on the functions of the performance tests. These 17 performance test group command files are listed in Section 5.5.1 on Page 5.15 of the User's Guide. The command files are written such that they can be easily divided into separate subgroup command files using a test editor.

By reviewing the "Application" performance test group command file, "ap.com", one can see it is made up of the following subgroups:

Artificial_intelligence
Avionics
Avl_tree
Cyclic_redundancy_check
Data_encryption_standard
Error_correcting_code
Filter
Integration
Kalman_filter
Lag_filter
Polynomial_coding_style
Simulation
Symmetric_deadzone
Symmetric_limiter
Trie

The discussion in the rest of this section presents an example on how to run the "Avionics" Subgroup of the "Application" Group. This was done on the AAAF VAX. Make a copy of the command file "ap.com" using a text editor. Delete all the lines in the copied file except those listed under the "Subgroup: avionics" comment header. Save the modified file under the new name "ap_av.com". The contents of this file is shown on the following pages.

```
$! ----------------------------------------------------
$! ---- Group: application
$! ------------ Subgroup: avionics
$! ----------------------------------------------------
$ set default [DAVIDSON.ACEC]
$ @zc_setli.com
$ set noon                               ! continue after errors
$ @zc_time1.com overhead
$ @zc_time2.com overhead
$ @zc_link1.com overhead
$ @zc_link2.com overhead
$ @zc_adaop.com    ap_avdum.ada                 ! begin dummy
$! delete          ap_avdum.ada;*               ! end dummy
$! ----------------------------------------------------
$ @zc_cpyop.com          ! copy params    ! begin main ap_avm01
$ @zg_incld.com    ap_av01_.inc  ap_av01_.ada      ! begin test
$ @zc_cmpop.com    ap_av01_.ada
$! delete          ap_av01_.inc;*
$ delete           ap_av01_.ada;*                ! end test
$ @zg_incld.com    ap_avm01.inc  ap_avm01.ada    ! main program
$ @zc_cmpop.com    ap_avm01.ada
$ @zc_link.com     ap_avm01
$ run              ap_avm01.exe
$ delete           ap_avm01.ada;*
$! delete          ap_avm01.inc;*
$! delete          ap_avm01.exe;*
$ @zc_delsb.com    ap_avm01            ! ada lib deletes
$ @zc_delsb.com    ifpm_control.ifpm_init.ifpm_rotors_init
$ @zc_delsb.com    ifpm_control.ifpm_init
$ @zc_delsb.com    ifpm_control.ifpm_rotors.ifpm_rotors_monitor
$ @zc_delsb.com    ifpm_control.ifpm_rotors
$ @zc_delsb.com    ifpm_control.ifpm_io
$ @zc_delsb.com    ifpm_control
$ @zc_delsb.com    nair.naircvt
$ @zc_delsb.com    nair
$ @zc_delsb.com    nscn.nscnvd
$ @zc_delsb.com    nscn
$ @zc_delsb.com    nutm.nutmcon
$ @zc_delsb.com    nutm
$ @zc_delsb.com    ifpm_constant
$ @zc_delsb.com    ifpm_status
$ @zc_delsb.com    ifpm_flags
$ @zc_delsb.com    nairini
$ @zc_delsb.com    nscni
$ @zc_delsb.com    nutmini
$ @zc_delsb.com    fmod
$ @zc_delsb.com    sin
$ @zc_delsb.com    cos
$ @zc_delsb.com    sgn
$ @zc_delsb.com    asum
$ @zc_delsb.com    atan2
$ @zc_delsb.com    ap_av01             ! end main ap_avm01
$ @zc_cpyop.com       ! copy params    ! begin main ap_avm02
$ @zg_incld.com    ap_av02_.inc  ap_av02_.ada      ! begin test
$ @zc_cmpop.com    ap_av02_.ada
$! delete          ap_av02_.inc;*
$ delete           ap_av02_.ada;*                  ! end test
$ @zg_incld.com    ap_avm02.inc  ap_avm02.ada    ! main program
$ @zc_cmpop.com    ap_avm02.ada
$ @zc_link.com     ap_avm02
$ run              ap_avm02.exe
$ delete           ap_avm02.ada;*
$! delete          ap_avm02.inc;*
$! delete          ap_avm02.exe;*
$ @zc_delsb.com    ap_avm02            ! ada lib deletes
$ @zc_delsb.com    ew_calculate_gates
$ @zc_delsb.com    compute_jitter_locklevel
$ @zc_delsb.com    ew_data_base
$ @zc_delsb.com    ew_pulse_assignment_logic
$ @zc_delsb.com    ew_timer
$ @zc_delsb.com    ew_track_file_update
$ @zc_delsb.com    ew_fifo
$ @zc_delsb.com    interrupt
$ @zc_delsb.com    read_out
$ @zc_delsb.com    z80_input_fifo
$ @zc_delsb.com    tracker_executive.initialize
$ @zc_delsb.com    tracker_executive.handle_intercept_message.-
acquisition
$ @zc_delsb.com    tracker_executive.handle_intercept_message.-
track_file_update
$ @zc_delsb.com    tracker_executive.handle_intercept_message
$ @zc_delsb.com    tracker_executive.handle_update_message.-
trailing_edge_process.drop_track
$ @zc_delsb.com    tracker_executive.handle_update_message.-
trailing_edge_process.new_hit_error_compensation
$ @zc_delsb.com    tracker_executive.handle_update_message.-
trailing_edge_process
$ @zc_delsb.com    tracker_executive.handle_update_message
$ @zc_delsb.com    tracker_executive
$ @zc_delsb.com    ap_av_ew            ! end main ap_avm02
$ @zc_cpyck.com       ! copy params    ! begin main ap_avm03
$ @zg_incld.com    ap_av03_.inc  ap_av03_.ada    ! begin test
$ @zc_cmpck.com    ap_av03_.ada
$! delete          ap_av03_.inc;*
$ delete           ap_av03_.ada;*                ! end test
$ @zg_incld.com    ap_avm03.inc  ap_avm03.ada    ! main program
$ @zc_cmpck.com    ap_avm03.ada
$ @zc_link.com     ap_avm03
$ run              ap_avm03.exe
$ delete           ap_avm03.ada;*
```

```
$! delete           ap_avm03.inc;*
$! delete           ap_avm03.exe;*
$ @zc_delsb.com     ap_avm03               ! ada lib deletes
$ @zc_delsb.com     ap_av03                ! end main ap_avm03
```

As shown on the previous pages, another required modification is to change the fifth line of this file from:

```
        $ set default acec_working_directory
```

to:

```
        $ set default [davidson.acec]
```

No other modifications to file "ap_av.com" are required.

2. Step 2 - Execute the Subgroup Command File

Execute file "ap_av.com" as a batch file with the results written to an output file. This output file will later be used as an input file by the ACEC Program Condense. Enter the VAX command:

```
  submit/notify/log_file=[davidson.acec]ap_av.log ap_av.com
```

This batch job will complete execution in about 8 minutes. After the terminal gives notification the batch job is completed, verify the file "ap_av.log" was created in the default directory. View the end of this file with an editor to verify no errors occurred. (Because it is rather large, printing this file is not recommended.)

3. Step 3 - Modify File "za_cosys.txt"

Using a text editor, modify the first occurrence of the lines that begin with ">system_name ..." to the following:

```
    >system_name                     := VAX
    >execution_log                   := ap_av.log
    >execution_condensed             :=
    >compilation_log                 := ap_av.log
    >compilation_condensed           :=
```

Delete everything which follow the lines shown above from file "za_cosys.txt".

4. Step 4 - Execute Program Condense

Execute the ACEC Program Condense through the Menu Program. Begin by entering the VAX Command:

```
        run menu
```

The Main Menu should be displayed. Enter:

```
    a,next          <--- to choose CONDENSE
    a,next          <--- to choose VAX
    e,next          <--- to create a new data base
    a,do            <--- to run Condense immediately
    quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
        file "ap_av.e00"
        file "ap_av.c00"
        file "za_coerr.txt"
```

These are the default names for these files. The name for the two data base files, "ap_av", is the same as the name used for the input file for the compilation and execution time analysis, "ap_av.log". The execution time data base file extension defaults to ".e00"; the compilation data base file extension defaults to ".c00". The digits "00" in the file extensions are incremented each time a new version of these files is created.

These files can be viewed with a text editor or printed. Review these files for error indications.

5. Step 5 - Execute the Single System Analysis (SSA)

Execute the ACEC Program Single System Analysis (SSA) through the Menu Program. Begin by entering the VAX Command:

```
        run menu
```

The Main Menu should be displayed. Enter:

```
    c,next          <--- to choose SINGLE_SYSTEM_ANALYSIS
    a,next          <--- to choose VAX
    d,next          <--- to request all reports
    a,do            <--- to run the analysis immediately
    quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
        file "vax.toc"
        file "vax.rep"
        file "vax.mis"
```

The main report is in file "vax.rep". File "vax.toc" is the table of contents for the main report. The page length of these reports is 64 lines per page and cannot be changed (ACEC User's Guide, Section 8.6, Page 8.43). These reports contain a footer line on Line 64 and if they are sent to a printer set for a maximum limit of less than 64 lines per page, the reports will not print properly. The footer line for each page is printed on a separate sheet following each regular report page. I experienced this problem on the AAAF VAX. The AAAF VAX System Manager provided me a special print utility command file, "dlp.com", to temporarily reset the printer limit for maximum-lines-per-page to 64. The reports then printed properly.

File "vax.mis" is a file of missing test analysis; it is always large (even when the full group is analyzed). Printing this file is not recommended but it can be viewed with a text editor.

These are the default names for these files. The file name "vax" is determined by the "system_name" specified by the user in file "za_cosys.txt". The default file extensions are determined by the type of report written to the file as explained in the above paragraph.

6. Step 6 - Modify File "za_cosys.txt" for Comparative Analysis

The Comparative Analysis (CA) needs data bases from at least two systems to execute. If only one target system is available, Comparative Analysis can still be executed by using the generic data bases provided on the ACEC Distribution Tape in the files, "za_smple.dbc" and "za_smple.dbs". To make use of these data base files, the files must be copied to the ACEC default directory. Also, the file "za_cosys.txt" MUST be modified by adding lines for the generic system "compare" as shown below:

```
    >system_name                       := VAX
    >execution_log                     := ap_av.log
    >execution_condensed               := ap_av.e00
    >compilation_log                   := ap_av.log
    >compilation_condensed             := ap_av.c00

    >system_name                       := compare
    >execution_log                     :=
    >execution_condensed               := za_smple.dbs
    >compilation_log                   :=
    >compilation_condensed            := za_smple.dbc
```

7. Step 7 - Execute the Comparative Analysis (CA) Program

Execute the ACEC Program Comparative Analysis (CA) through the Menu Program. Begin by entering the VAX Command:

```
        run menu
```

The Main Menu should be displayed Enter:

```
    b,next          <--- to choose COMPARATIVE_ANALYSIS
    c,next          <--- to choose All Systems
    f,next          <--- to request All Metrics
    a,next          <--- to choose the APPLICATION group
    b,f,next        <--- to request FULL_REPORT & SUMMARY_REPORT
                         and have all reports written to a
                         single file, "compar00.rpt"
    a,do            <--- to run the analysis immediately
    quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
        file "compar00.rpt"
        file "za_cadb.txt"
        file "za_coerr.txt"
```

The report is contained in file "compar00.rpt", as requested during the execution of the Menu Program. This file should also be printed on a printer will the maximum-lines-per-page limit set for 64. On the AAAF VAX, I found it necessary to use the "dlp.com" command file utility to print this file, as I did for the SSA reports (see Step 5 of this Section).

File "za_cadb.txt" is the summary data base for CA and is explained in Section 8.4.2 on Page 8.34 of the User's Guide. File "za_coerr.txt" is created every time an analysis is executed using the Menu Program and includes execution errors (if any).

In this case, file "za_coerr.txt" may contain a warning that the data bases for System "VAX" and System "compare" do not match in size. This warning appears because the data bases for System "compare" contain data for all the test groups whereas the data bases for System "VAX" contain data for only the subgroup "Avionics." The User's Guide cautions in section 8.4.2 at the bottom of Page 8.34, that CA will start a new database when an inconsistency is discovered. This may result in the loss of previous results unless the name of the CA database file, "za_cadb.txt", is changed in the System Names file, "za_cosys.txt".

Section 5

RUNNING PERFORMANCE TESTS FOR A FULL GROUP

1. Step 1 - Modify a Performance Group Command File

It is easier to prepare ACEC for analyses on one of the performance test groups (listed in the User's Guide, Section 5.5.1, Page 5.15) than on a subgroup of a performance test group as discussed in Section 4 of this report. Of course, computer processing time will increase significantly and the size of the analysis reports produced will be much greater.

The discussion in the rest of this section presents an example on how to run the "Application" performance test group. This was done on the AAAF VAX. In this case, the entire command file "ap.com" is used. Fourteen more modifications are required to change each occurrence of the line:

```
$ set default acec_working_directory
```

to:

```
$ set default [davidson.acec]
```

No other modifications to file "ap.com" are required.

2. Step 2 - Execute the Subgroup Command File

Execute file "ap.com" as a batch file with the results written to an output file. This output file will later be used as an input file by the ACEC Program Condense. Enter the VAX command:

```
submit/notify/log_file=[davidson.acec]ap.log ap.com
```

This batch job will complete execution in about **1.5 hours**. After the terminal gives notification the batch job is completed, verify the file "ap.log" was created in the default directory. View the end of this file with an editor to verify no errors occurred. (Because it is VERY LARGE -1400 lines-, printing this file is not recommended.)

3. Step 3 - Modify File "za_cosys.txt"

Using a text editor, modify the first occurrence of the lines that begin with ">system_name ..." to the following:

```
>system_name                        := VAX
>execution_log                      := ap.log
>execution_condensed                :=
>compilation_log                    := ap.log
>compilation_condensed              :=
```

4. Step 4 - Execute Program Condense

Execute the ACEC Program Condense through the Menu Program. Begin by entering the VAX Command:

```
run menu
```

The Main Menu should be displayed. Enter:

```
a,next          <--- to choose CONDENSE
a,next          <--- to choose VAX
e,next          <--- to create a new data base
a,do            <--- to run Condense immediately
quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
file "ap.e00"
file "ap.c00"
file "za_coerr.txt"
```

These are the default names for these files. The name for the two data base files, "ap", is the same as the name used for the input file for the compilation and execution time analysis, "ap.log". The execution time data base file extension defaults to ".e00"; the compilation data base file extension defaults to ".c00". The digits "00" in the file extensions are incremented each time a new version of these files are created.

These files can be viewed with a text editor or printed. Review these files for error indications.

5. Step 5 - Execute the Single System Analysis (SSA)

Execute the ACEC Program Single System Analysis (SSA) through the Menu Program. Begin by entering the VAX Command:

```
run menu
```

The Main Menu should be displayed. Enter:

```
c,next          <--- to choose SINGLE_SYSTEM_ANALYSIS
a,next          <--- to choose VAX
d,next          <--- to request all reports
a,do            <--- to run the analysis immediately
quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
file "vax.toc"
file "vax.rep"
file "vax.mis"
```

The main report is in file "vax.rep". File "vax.toc" is the table of contents for the main report. Print this file on a printer with the maximum-lines-per-page set for at least 64. Refer to the discussion in Step 5 of Section 4 of this report.

File "vax.mis" is a file of missing test analysis; it is always large (even when the full group is analyzed). Printing this file is not recommended but it can be viewed with a text editor.

These are the default names for these files. The file name "vax" is determined by the "system_name" specified by the user in file "za_cosys.txt". The default file extensions are determined by the type of report written to the file as explained in the above paragraph.

6. Step 6 - Modify File "za_cosys.txt" for Comparative Analysis

The Comparative Analysis (CA) needs data bases from at least 2 systems to execute. If only one target system is available, Comparative Analysis can still be executed by using the generic data bases provided on the ACEC Distribution Tape in the files, "za_smple.dbc" and "za_smple.dbs". To make use of these data base files, the files must be copied to the ACEC default directory (unless this was already done for Step 6 in Section 4). Also, the file "za_cosys.txt" MUST be modified by adding lines for the generic system "compare" (unless this was already done for Step 6 in Section 4) as shown below:

```
>system_name                       := VAX
>execution_log                     := ap.log
>execution_condensed               := ap.e00
>compilation_log                   := ap.log
>compilation_condensed             := ap.c00

>system_name                       := compare
```

```
>execution_log                    :=
>execution_condensed              := za_smple.dbs
>compilation_log                  :=
>compilation_condensed            := za_smple.dbc
```

The system_name "compare" in file "za_cosys.txt" must be used because this is the name that appears for the system name in the first line of files "za_smple.dbc" and "za_smple.dbc". CA requires this agreement for system names or execution will be terminated.

7. Step 7 - Execute the Comparative Analysis (CA) Program

Execute the ACEC Program Comparative Analysis (CA) through the Menu Program. Begin by entering the VAX Command:

```
run menu
```

The Main Menu should be displayed. Enter:

```
b,next          <--- to choose COMPARATIVE_ANALYSIS
c,next          <--- to choose All Systems
f,next          <--- to request All Metrics
a,next          <--- to choose the APPLICATION group
b,f,next        <--- to request FULL_REPORT & SUMMARY_REPORT
                     and have all reports written to a
                     single file, "compar00.rpt"
a,do            <--- to run the analysis immediately
quit            <--- to exit the ACEC Menu Program
```

Verify the following files were created:

```
file "compar00.rpt"
file "za_cadb.txt"
file "za_coerr.txt"
```

The report is contained in file "compar00.rpt", as requested during the execution of the Menu Program or in file "compar01.rpt" if file "compar00.rpt" was previously created in Section 4. This file should also be printed on a printer will the maximum-lines-per-page limit set for 64. On the AAAF VAX, I found it necessary to use the "dlp.com" command file utility to print this file, as I did for the SSA reports (see Step 5 of Section 4).

File "za_cadb.txt" is the summary data base for CA and is explained in Section 8.4.2 on Page 8.34 of the User's Guide. File "za_coerr.txt" is created every time an analysis is executed using the Menu Program and includes execution errors (if any).

In this case, file "za_coerr.txt" may contain a warning that the data bases for System "VAX" and System "compare" do not match in size. This warning appears because the data bases for System "compare" contain data for all the test groups whereas the data bases for System "VAX" contain data for only the group "APPLICATION." The User's Guide cautions in Section 8.4.2 at the bottom of Page 8.34, that CA will start a new database when an inconsistency is discovered. This may result in the loss of previous results unless the name of the CA database file, "za_cadb.txt", is changed in the System Names file, "za_cosys.txt".

References

1. "Ada Compiler Evaluation Capability (ACEC) Version 3.0, Now Available," Ada Information Clearing house Newsletter (The Ada Joint Program Office), Vol. IX, No. 4, December 1991, pp. 9 and 11.

2. Sara Decker, Tom Leavitt, Ada Compiler Evaluation Capability (ACEC), Technical Operating Report (TOR), User's Guide Release 3.0, Boeing Defense and Space Group, Product Support Division, P.O. Box 7730, Wichita, Kansas, Document No. D500-12564-1, December 17, 1991.

U.S. GOVERNMENT PRINTING OFFICE 750-113